# History on Demand (HoD) - Direct - v3

Domain Portfolio: Conditions | Domain: Historical | Usage Classification: **Limited Availability**

**Geography:** Global **Attribution Required:** No **Attribution Requirements:** N/A

## Overview

### The Intent of the History on Demand APIs

The History on Demand (HoD) APIs provide access to global, historical weather data. Historical weather data is prevalent in trend analysis and the training of analytical models pertaining to energy, agriculture, insurance and many other industries. These APIs serve the same data, but allow you to collect that data the way that works best for you. Each has its own style of interaction, allowable query types, delivery mode, and restrictions. Select the API that suits your usage style, desired data format and delivery mechanism. **HoD Direct** (this API) is a synchronous approach, and **HoD Archive** (the other API) is asynchronous. Information common to both is centralized within one document, which is linked, where necessary, from within each respective API document.

### The API

**HoD Direct** is a synchronous API. Users specify the geospatial and the temporal facets of the query in the URL, and the data is returned in the response. See the Data Layers section in the HoD Common Information document for details on data layers served by the HoD APIs. The API implements paging, so that no single request takes excessively long to come back. It also allows for the specification of a geolocation in multiple formats, explained below. The format of the data in a successful response is dictated by the **`format`** specified in the request. The format of all other responses (request validation errors, etc.) is json.

### Caching and Content Type

This API does not use response caching and is not sensitive to headers (e.g. Accept, Accept-Encoding). Response type is dictated by the `format` parameter..

### Special Considerations Regarding Your Use of this API

- We have prepared a Postman Collection for your convenience in interacting with the API. Feel free to download it, and customize it for your needs.
- If this is your first time accessing the API, we strongly recommend you review the Example API Interaction for a walkthrough in how to use **HoD Direct**.
- For a complete understanding of the rules and interpretation on date and date/time expressions sent to the API, see the Date / Time Inputs and Interpretation section in the HoD Common Information document.
- There are several ways to specify a location you wish to query. See Parameters Specific to Location section of this document.
- Calls to this API will record usage for your API key, viewable through the /usage endpoint. See the HoD Usage section in the HoD Common Information document.
- **This API implements paging**, which is explained in the Paging section.
- For API Usage calculations, see Usage Calculation in the HoD Common Information document.
- This API is supplemented by the HoD Utility API. Details can be found in the corresponding section of the HoD Common information document.

## Atomic Endpoints

| |
|---|
| v3/wx/hod/r1/direct |
| v3/wx/hod/r1/usage |

## URL Construction

**Request Data**
**Method: GET**
**Required Request Parameters:** apiKey, startDateTime, endDateTime, format, units, [some location specification]
**Optional Request Parameters:** geocode, wkt, icaoCode, iataCode, postalKey, weighted, pageNumber, compact
https://api.weather.com/v3/wx/hod/r1/direct?geocode=25.807199,-80.122034&startDateTime=<**startDateTime**>&endDateTime=<**endDateTime**>&format=<**format**>&units=<**units**>&apiKey=**yourApiKey**

https://api.weather.com/v3/wx/hod/r1/direct?geocode=25.807199,-80.122034&startDateTime=2020-05-01T00Z&endDateTime=2020-05-15T00Z&format=json&units=e&apiKey=**yourApiKey**
https://api.weather.com/v3/wx/hod/r1/direct?geocode=25.807199,-80.122034&startDateTime=2020-05-01T00Z&endDateTime=2020-05-15T00Z&format=json&units=e&pageNumber=3&compact=false&apiKey=**yourApiKey**
https://api.weather.com/v3/wx/hod/r1/direct?geocode=25.807199,-80.122034&weighted=true&startDateTime=2020-05-01T00Z&endDateTime=2020-05-15T00Z&format=json&units=e&pageNumber=3&apiKey=**yourApiKey**
https://api.weather.com/v3/wx/hod/r1/direct?geocode=25.807199,-80.122034&products=Temperature,
EvapoTranspiration&startDateTime=2020-05-01T00Z&endDateTime=2020-05-15T00Z&format=json&units=e&pageNumber=3&apiKey=**yourApiKey**
https://api.weather.com/v3/wx/hod/r1/direct?geocode=25.807199,-80.122034&products=all&startDateTime=2020-05-01T00Z&endDateTime=2020-05-15T00Z&format=json&units=e&pageNumber=3&apiKey=**yourApiKey**
https://api.weather.com/v3/wx/hod/r1/direct?wkt=POINT%20(25.807199%20-80.122034)&startDateTime=2020-05-01T00Z&endDateTime=2020-05-15T00Z&format=json&units=e&apiKey=**yourApiKey**
https://api.weather.com/v3/wx/hod/r1/direct?icaoCode=VQPR&startDateTime=2020-05-01T00Z&endDateTime=2020-05-15T00Z&format=json&units=e&apiKey=**yourApiKey**
https://api.weather.com/v3/wx/hod/r1/direct?iataCode=CGH&startDateTime=2020-05-01T00Z&endDateTime=2020-05-15T00Z&format=json&units=e&apiKey=**yourApiKey**
https://api.weather.com/v3/wx/hod/r1/direct?postalKey=100-6210:JP&startDateTime=2020-05-01T00Z&endDateTime=2020-05-15T00Z&format=json&units=e&apiKey=**yourApiKey**

For **HoD Usage** URL construction, see the HoD Usage section in the HoD Common Information document.

For **HoD Utility** URL construction, see the HoD Utility section in the HoD Common Information document.

## Valid Parameter Definitions

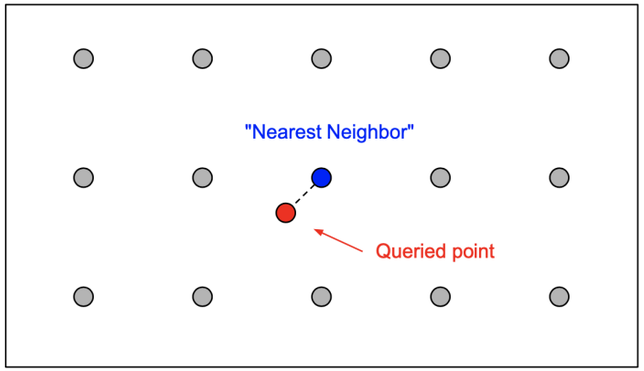
GET /v3/wx/hod/r1/direct

| Parameter | Description | Example |
|---|---|---|
| startDateTime | The beginning of the analysis period. Must be before endDateTime. **Inclusive.** See Date / Time Inputs and Interpretation | **startDateTime**=20200501T00Z |
| endDateTime | The end of the analysis period. Must be after startDateTime. **Exclusive.** See Date / Time Inputs and Interpretation | **endDateTime**=20200515T00Z |
| format | The desired format of the resulting data: **json** or **csv** | **format**=json |
| units | Unit system in which the response should be expressed: Must be one of 'e', 'english', 'm', 'metric', 's', or 'si | **units**=s |
| compact | Boolean: whether you would prefer json response to be in "compact array" format. If false, natural JSON format is used. Default = true. | **compact**=false |
| products | The case-insensitive list of products whose data should be included in the response<br>Omission results in all gCOD products<br>“all” results in all available weather products.<br>See special note in "weighted" parameter description | **products**=temperature,evapotranspiration<br>**products**=all |
| **Parameters specific to Location** | | |
| geocode | comma-separated "`latitude,longitude`" expression | **geocode**=`-27.125736,-109.276746` (no spaces) |
| wkt | Valid Well-Known-Text spatial geometry representation<br>See Location Specification section in the HoD Common Information document<br>Supports POINT, BBOX, and POLYGON | **wkt**=`POINT (40.748582 -73.986034)`<br>(must be URL encoded: spaces = %20 or + ) |
| ICAO code | see ICAO Airport Code | **icaoCode**=VQPR |
| IATA code | see IATA Airport Code | **iataCode**=CGH |
| postal Key | in the format [postal Code]:[2-digit country code] | **postalKey**=100-6210:JP |
| weighted | Exclusively used with point queries, indicates that the desired data should be an interpolation of the values from the four nearest neighboring grid points, rather than the values of the single nearest neighboring grid point. the default value is `false`.<br>**Special Note about weighted average and choosing specific products in the request**:<br>specifying `weighted=true`, while also explicitly specifying `products` upon which weighted averaging cannot be performed, will result in a validation error. Example: `...&weighted=true&products=iconCode` (which is an enumerated value) | **weighted**=true |
| **Parameters specific to paging** | | |
| pageNumber | Integer value indicating the page of data desired as the response to the call. | **pageNumber=3** |

For **HoD Usage** Valid Parameter Definitions, see the HoD Usage section in the HoD Common Information document.

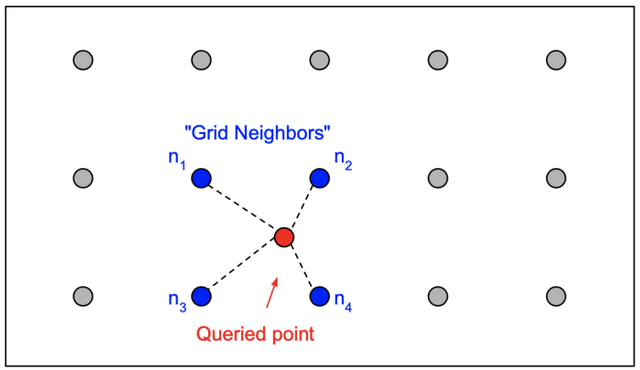
For **HoD Utility** Valid Parameter Definitions, see the HoD Utility section in the HoD Common Information document.

## Point Query Types




### Nearest Neighbor

The nearest neighbor query is the default point query. Any point query which does not include the "`weighted=true`" parameter, is a nearest neighbor query. The data returned is said to be "representative of" the queried point. In the result set for a nearest neighbor query, there is differentiation between the "`requestedLatitude / requestedLongitude`" and the [nearest neighbor] "`latitude / longitude`" to highlight where the representative data was drawn from. It is also important to note that if the point queried is specified via `postalKey`, `icaoCode` or `iataCode`, the expression is first sent to Location Services to resolve it to the geolocation with which it is associated. Then *that point* is resolved to a nearest neighboring grid point.




### Weighted Average

The weighted average query is a specialized point query in which the values are calculated from the values of the four nearest neighboring grid points. This is true for each of the weather aspects (Temperature, Wind Speed, etc.) stored at the grid locations. There are three exceptions: `Driving Difficulty Index`, `Icon Code`, and `Icon Code Extended`. These values will always be null for a point queried using a weighted average query. These are enumerated rather than scalar values, therefore averaging them is meaningless. Please note the special relationship between the `products` parameter and weighted queries. Explicitly choosing `products` to which the weighted average algorithm cannot be applied will result in a validation error. If the `products` parameter is either omitted or is set to "`all`", the API will behave normally, simply omitting the irrelevant products as described.

The math behind the Inverse Distance Weighted Average algorithm is as follows:

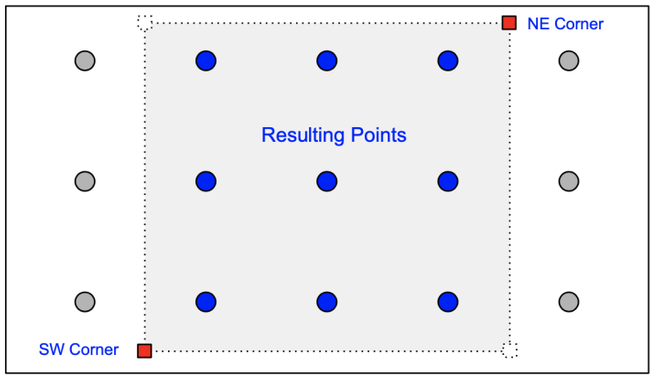
$$\frac{\sum \text{weatherValue}[n] \;/\; \text{distance}[n]}{\sum 1 \;/\; \text{distance}[n]}$$

That is, using Temperature as an example, the sum of each neighbor's Temperature value over their distance (in meters) from the queried point, all over the sum of the reciprocal (multiplicative inverse) of each neighbor's distance from the queried point. In the response to a weighted point query, you will notice that the "`requestedLatitude / requestedLongitude`" and the "`latitude / longitude`" will be the same coordinate pair, since the values returned have been mathematically calculated to represent that point. One final technical piece of information regards the elevation of the queried point, and the elevation variations of the neighbors whose values are used to calculate the average. Since the observations recorded for each grid point occur at different elevations, to truly normalize neighboring values, one would have to adjust those neighbors' values that are sensitive to elevation (e.g. Temperature, Pressure), such that each neighbor would be "raised" or "lowered" to the elevation of the queried point. Only then could they be considered mathematically equal participants in the average calculation. Superimposing a "Digital Elevation Model" on the gridded data in this way is a feature being investigated. This information is provided for meteorological consideration. Its significance will depend upon the variation in the terrain surrounding the queried point.

## Boundary Query Types

Boundary queries allow you to specify geospatial areas. The result of these queries will be the set of all grid points contained within the area specified. Edges are inclusive, meaning any grid point falling precisely on the border of a specified area will be considered "contained" within the area. Note that the projection of any specified geospatial area on the Earth will cause some minor variance in the precision of the determination of containment.

This type of query can by nature produce results that are quite large. As such the API is configured to curtail the workload any one caller can produce. These limitations are fairly generous and should adequately suit any focused query. For much larger data sets, consider using HoD Archive. The limitation for any bounding box, or for the rectangle said to "minimally contain" a specified polygon, has been initially set to 2 million $km^2$ . This setting is subject to change at any time. Beyond that geospatial limitation, resulting data sets have been initially limited to 100,000 records, or 10 pages of data (page size has been initially set to 10,000 records). When designing your boundary query, consider that the total record count is **`[# grid points contained in boundary] x [# hours queried]`**. Also note that the results will come back in point-first order, meaning all the hours for the first point, then all the hours for the next point, and so on. Validation messages will be very helpful when constructing and running boundary queries.

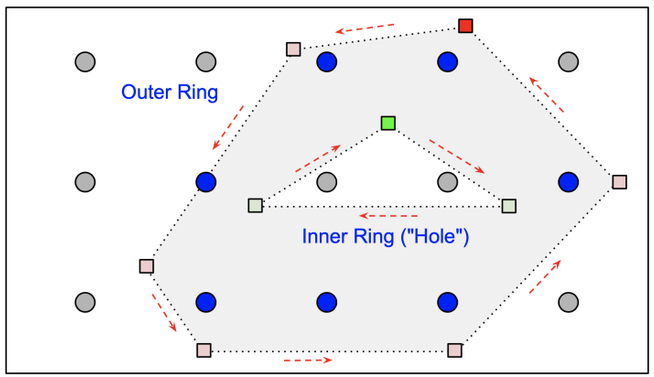



### Bounding Box

Bounding Box ("bbox") queries are a simple way to capture a [projected] "rectangular" area on the surface of the Earth.

**`BBOX ((SW_lat SW_lon), (NE_lat NE_lon))`**

In this example, the BBOX expression would specify the southwestern red point, then the northwestern red point, resulting in 9 grid points for each hour queried.




### Polygon

Polygon queries are more complicated to specify, but can be very useful to query oddly shaped areas. Polygons should be specified using the Geoshape WKT for Geometries Standard. This details important aspects of how to define a polygon so that it is properly interpreted. At left, we have an outer ring starting (and ending) with the red square, listing its vertices in counter-clockwise order. The inner ring starts (and ends) with the green square, listing its vertices in clockwise order.

One thing to remember when designing a polygon query is the grid data density. A poly whose detail is more granular than the density of the data could entirely miss available nearby grid points.The area limitation for polygons is implemented in HoD Direct as follows: first, the API checks the area of a rectangle just barely big enough to contain the outer ring, then it checks the product of the number of data points the poly encases and the number of hours queried.

**`POLYGON ((outer_pt1_lat outer_pt1_lon, outer_pt2_lat outer_pt2_lon, . . . , outer_pt1_lat outer_pt1_lon), (inner_pt1_lat inner_pt1_lon, inner_pt2_lat inner_pt2_lon, . . . , inner_pt1_lat inner_pt1_lon))`**

# VERY IMPORTANT

The HoD Direct API is configured to return data in pages if the entire dataset exceeds the configured page size. **The page size is internally managed and may change at any time** based upon performance observations. This means that if interactions with the API do not accommodate paging, a query that once returned the entirety of the desired data set might suddenly only return a page of that data set. It is therefore advisable to integrate paging into programmatic interactions with the API.

Any call to the API which will require paged calls to fully retrieve the data will include a **`Next-Page-Number`** header, with an integer page number, along with the response. That integer value should be supplied in the next request, in a request parameter called **`pageNumber`**. Any nonsensical values (negative page numbers, page numbers beyond the last page of a result) will be dealt with as follows: negatives are changed to 0, values larger than the last page number will result in a **`204 - No Content`**.

Pseudo code for programmatically accommodating paging is as follows:

```
set up query (dates, point, format, etc.)

make first call
store result locally

while (next-page-number header is present in response)

  make same call with param pageNumber=[val from response header]
  append to local result

done
```

## Data Elements & Definitions

HoD Direct now serves weather products from multiple datasets.  Currently the datasets are Gridded Currents on Demand (gCOD) and Agriculture and Energy (AgE).
See the Data Elements & Definitions section in the HoD Common Information document for details on datasets available in the response.
There are a few important points regarding requests and responses involving multiple datasets:

- The **`products`** request parameter can be used to customize the content in the response.
    - omitting the **`products`** parameter will yield a response including only the gCOD products, which has been the case since the API's inception.
    - **`products=all`** will yield a response including all weather products from all datasets.  This is the best way to discover the names of the products you can specify.
    - those product identifiers can then be used in a case-insensitive expression. E.g. **`products=temperature,temperatureDewpoint,evapotranspiration,globalHorizontalIrradiance`**
- The timestamp on the hourly data will react to which products are requested.
    - All data served by HoD Direct is hourly data, so any given weather product's hourly value is said to be representative of that hour, even though the production timestamps of the data may differ from dataset to dataset.
    - gCOD's production timestamp is always `":20"` minutes after the hour
    - AgE's production timestamp is always `":00"` minutes after the hour
    - Therefore, if the request only includes gCOD values, the hourly timestamps will look like: `"validTimeUtc": "2022-11-01T21:20:00+0000"`
    - If any AgE product is included in the response, the hourly timestamps will default to `":00"` minutes: `"validTimeUtc": "2022-11-01T21:00:00+0000"`

For **HoD Usage** Data Elements & Definitions, see the HoD Usage section in the HoD Common Information document.

For **HoD Utility** Data Elements & Definitions, see the HoD Utility section in the HoD Common Information document.

## Validation Messages

Validation (400 BAD REQUEST) messages are very simple, clear and informative in this API, and all known validation issues are reported on each request.
For example, if you were to issue a query with just the base URL and a valid API Key, but no other required fields, you would get a response similar to the following:


```
{
    "status": 400,
    "errors": [
        {
            "fieldName": "startDateTime",
            "message": "Missing required field. Must be of the form yyyy-MM-dd'T'HH:mm:ssZ or yyyyMMdd'T'HHmmssZ."
        },
        {
            "fieldName": "endDateTime",
            "message": "Missing required field. Must be of the form yyyy-MM-dd'T'HH:mm:ssZ or yyyyMMdd'T'HHmmssZ."
        },
        {
            "fieldName": "format",
            "message": "Missing required field. Must be 'csv' or 'json'."
        },
        {
            "fieldName": "units",
            "message": "Missing required field. Must be one of 'e', 'english', 'm', 'metric', 's', or 'si'."
        },
        {
            "fieldName": "geocode",
            "message": "Missing required field. Must be of the form latitude,longitude with values in decimal degrees (e.g. 33.916,-84.341)."
        }
    ]
}
```

## Example API Interaction

In this section are some simple examples of the ways you can interact with the API.  Note that the **apiKey** parameter would be required in all cases.

### Request Data - all gCOD vars only

| **REQUEST**<br><br>**GET /v3/wx/hod/r1/direct?geocode=25.807199,-80.122034&startDateTime=2020-05-01T00Z &endDateTime=2020-05-15T00Z&format=json&units=e&compact=false&apiKey =** | **RESPONSE Body**<br><br>(fake data, one partial object in a list) | **RESPONSE Headers** |
|---|---|---|
| | [<br>{<br>"requestedLatitude": 25.807199,<br>"requestedLongitude": -80.122034,<br>"latitude": 25.824,<br>"longitude": -80.112,<br>"gridpointId": "value",<br>"validTimeUtc": "2021-04-05T12:14:04+0000",<br>"drivingDifficultyIndex": 0,<br>"iconCode": 31,<br>"iconCodeExtended": 1800,<br>"precip1Hour": 0.0194785,<br>"precip6Hour": 0.59179074,<br>. . . [omitted content] . . .<br>"temperature": 0.8188712,<br>"temperatureChange24Hour": 0.80135256,<br>"temperatureMax24Hour": 0.21653223,<br>"temperatureMin24Hour": 0.06830543,<br>"temperatureDewPoint": 0.032562375,<br>"temperatureFeelsLike": 0.41878968,<br>"uvIndex": -92,<br>"visibility": 0.11596674,<br>"windDirection": 0.4087097,<br>"windGust": 0.05093181,<br>"windSpeed": 0.77035826<br>},<br>. . .<br>] | **Request-Id**=[ID assigned to your query]<br>This header specifies the Request Id.  It can be used for inquiries.<br><br>**Content-Disposition**=attachment;filename=hod-direct-[Request ID]<br>This header indicates that the response will be an attachment to be downloaded and saved locally, rather than displayed inline.  If you are using tools like Curl, Postman, or a programming language, you can control how this is handled. Most browsers will honor the header and download the response as an attachment.<br><br>**Content-Type**=[content type]<br>This header reflects the response format chosen by the caller.<br>("application/json", "text/csv", etc.)<br><br>**Next-Page-Number**=[integer]<br>If the query result is paginated, and if there is a next page, this header specifies integer page number to be sent in request parameter **pageNumber** in order to retrieve the next page of data.  If either of those conditions is not true, the header is not included in the response. |

## Request Data - all available data

| REQUEST | RESPONSE Body | RESPONSE Headers |
|---|---|---|
| **GET /v3/wx/hod/r1/direct?geocode=41.0,&startDateTime=2020-08-01T18Z&endDateTime=2020-08-15T19Z&format=json&units=e&compact=false&products=all&apiKey=** | (fake data, one partial object in a list) | |
| Note the inclusion of the **products=all** parameter | same as gCOD only, except will include 2 AgE fields if query time range contains hours for which the AgE dataset is present<br>[<br>{<br>. . .<br>"evapotranspiration": 0.0,<br>"globalHorizontalIrradiance": 258.1,<br>. . .<br>] | same as previous |

## Request Data - only specific weather products across datasets

| REQUEST | RESPONSE Body | RESPONSE Headers |
|---|---|---|
| **GET https://api.weather.com/v3/wx/hod/r1/direct?geocode=41.0,&startDateTime=2020-08-15T18Z&endDateTime=2020-08-15T19Z&format=json&units=e&compact=false&products=temperature,temperatureDewpoint,evapotranspiration,globalHorizontalIrradiance&apiKey=** | (fake data, one partial object in a list) | |
| Customize your response by including only those fields you need.<br><br>The example specifies:<br>products=temperature,temperatureDewpoint,evapotranspiration,globalHorizontalIrradiance | [<br>{<br>"requestedLatitude": 41.0,<br>"requestedLongitude": 0.0,<br>"latitude": 40.989010989010985,<br>"longitude": -3.084749914705753E-5,<br>"gridpointId": "ezrcpvzfr32yx",<br>"validTimeUtc": "2020-08-15T18:00:00+0000",<br>"evapotranspiration": 0.0,<br>"globalHorizontalIrradiance": 194.9,<br>"temperature": 81.2,<br>"temperatureDewPoint": 65.0<br>}<br>] | same as previous |

For **HoD Usage** API Interaction, see the HoD Usage section in the HoD Usage section in the HoD Common Information document.

For **HoD Utility** API Interaction, see the HoD Utility section in the HoD Common Information document.